# Sitsang / PXA255 Evaluation Platform

## Linux User's Guide

***September 2003***

# History

| Date | Description | Owner |
|---|---|---|
| September, 2003 | Update Manual for Sitsang / PXA255 Linux Version. | Liu, Kent |
| February, 2003 | Initial official Manual of pxa250-gcc-2.95.3 version. | Alvin, Tang |
| | | |

Information in this document is provided in connection with Intel® products. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document. Except as provided in Intel's Terms and Conditions of Sale for such products, Intel assumes no liability whatsoever, and Intel disclaims any express or implied warranty, relating to sale and/or use of Intel products including liability or warranties relating to fitness for a particular purpose, merchantability, or infringement of any patent, copyright or other intellectual property right. Intel products are not intended for use in medical, life saving, or life sustaining applications.
Intel may make changes to specifications and product descriptions at any time, without notice.
Designers must not rely on the absence or characteristics of any features or instructions marked “reserved” or “undefined.” Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.
The Sitsang / PXA255 Evaluation Platform may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.
MPEG is an international standard for video compression/decompression promoted by ISO. Implementations of MPEG CODECs, or MPEG enabled platforms may require licenses from various entities, including Intel Corporation. Chips, Intel, Intel logo, Intel Inside, Intel Inside logo, Intel SpeedStep, Intel StrataFlash, Intel XScale, are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
*Other names and brands may be claimed as the property of others.

# Contents

1 Introduction ..... 1-1
1.1 Supported Drivers Summary ..... 1-1
1.2 Related Documents ..... 1-1
1.3 Document Content ..... 1-1
2 Quick Start ..... 2-1
2.1 Dependence ..... 2-1
2.1.1 Software Resources ..... 2-1
2.1.2 Hardware Resources ..... 2-2
2.2 Cable Connection ..... 2-2
2.3 Linux Loading Steps ..... 2-2
2.4 Network Configuration ..... 2-6
2.5 Configuring User's Development Environment for TFTP Downloading ..... 2-7
2.5.1 Configuring TFTP server for RedHat (Mandrake) Linux on host: ..... 2-7
2.5.2 Configuring Redboot on Sitsang board: ..... 2-7
2.6 Mounting NFS Partition and RAMFS Partition ..... 2-9
2.6.1 Mounting NFS partition ..... 2-9
2.6.2 Mounting RAMFS partition ..... 2-9
2.7 Loading QTOPIA Application ..... 2-9
2.8 USB Client ..... 2-10
2.8.1 Communicating with *Cat* ..... 2-10
2.8.2 Communicating with PPP ..... 2-11
2.9 USB Host ..... 2-11
2.10 AC97 ..... 2-12
2.10.1 Record via microphone ..... 2-12
2.10.2 Tune the volume ..... 2-12
2.10.3 Set the record source ..... 2-13
2.10.4 Change the A/D and D/A frequency ..... 2-13
2.11 CF Card ..... 2-13
2.12 Redboot Flash Partition Table ..... 2-13
2.12.1 Dual Flash Banks Scheme ..... 2-14
2.12.2 Single Flash Bank Scheme ..... 2-14
2.13 Accelerometer Sensor ..... 2-14
2.14 Backlight Control Interface ..... 2-14
2.15 IRDA ..... 2-15
2.16 Demo List ..... 2-15
2.17 Device File List ..... 2-15
3 Linux for Sitsang from Scratch ..... 3-1
3.1 Dependence ..... 3-1
3.2 Installing the Tool Chain ..... 3-1
3.3 Building Redboot from Source Code ..... 3-2
3.4 Building Linux Kernel from Source Code ..... 3-2
3.5 Creating A JFFS2 File System Image ..... 3-4

# Tables

Table 1: Linux Binary Packages List ..... 2-1
Table 2: Hardware List ..... 2-2
Table 3: Settings of the HyperTerminal or minicom ..... 2-2
Table 4: Frequency configuration based on the value of S7 ..... 2-6
Table 5: Module List for USB Client ..... 2-10
Table 6: Module List for USB Host ..... 2-11
Table 7: Mic Volume Setting List ..... 2-12
Table 8: Module List for CF Card ..... 2-13
Table 9: Two Flash Banks Scheme Partition Table of Redboot ..... 2-14
Table 10: Single Flash Bank Partition Table of Redboot ..... 2-14
Table 11: Demo List ..... 2-15
Table 12: Device File List ..... 2-15
Table 13: Linux Tool Chain and Source Codes List ..... 3-1
Table 14: Kernel Module List ..... 3-3

# 1 Introduction

This document describes the steps of installing Linux system on Sitsang board with binary release packages and the steps of building the binary packages from scratch. Linux distribution version for **Sitsang PXA255 Evaluation Platform** is Linux 2.4.19 B-1-2-1, which adds more drivers support than previous version and improves the performance of some drivers.

## 1.1 Supported Drivers Summary

- USB Client
- USB Host ISP1161
- CF Ethernet Card
- 802.11b Wireless Network CF Card
- Accelerometer Sensor
- Joystick & Soft button
- AC97 Recorder & Line In
- LCD Controller - 640x480
- LCD 16bpp display support, backlight control
- Touch Screen: ADS7846 support (intended to be used in conjunction with tslib from http://www.arm.linux.org.uk/cvs)
- Stereo Audio
- Product Ethernet support SMC 91C96
- IrDA
- 32-bit Flash EEPROM
- Real Time Clock (RTC)
- OS Timers
- Interrupt controller
- Memory Management Unit
- Serial Port (FFUART, BTUART)
- Memory controller

## 1.2 Related Documents

**Sitsang Linux / PXA255 Evaluation Platform User's guide**

## 1.3 Document Content

The following chapters are included in this document:

1. Introduction: Summary words for this document.
2. Quick Start: A manual for users to install and use Sitsang Linux on Sitsang / PXA255 evaluation Platform.
3. Linux for Sitsang from Scratch: Building every parts of Sitsang Linux from scratch packages.

# 2 Quick Start

In this document, the character '%' and '$' are the command prompts of Linux Operating System of Sitsang / PXA255 Evaluation Platform. It may be different from some other Linux system. 'Redboot>' is the command prompt of Redboot. Please don't be confused with the following command that user needs to input. All commands, which user needs to input are in ***Black and Italic*** font, such as "***cd /tmp***". All messages displayed on the screen automatically are in blue color and ***Italic*** font, such as "*RedBoot(tm) debug environment*".

This chapter contains:

Section 2.1 — Dependence
Section 2.2 — Cable Connection
Section 2.3 — Linux Loading Steps
Section 2.4 — Network Configuration
Section 2.5 — Configuring User's Development Environment for TFTP Download
Section 2.6 — Mounting NFS Partition and RAMFS Partition
Section 2.7 — Loading QTOPIA Application
Section 2.8 — USB Client
Section 2.9 — USB Host
Section 2.10 — AC97
Section 2.11 — CF Card
Section 2.12 — Redboot Flash Partition Table
Section 2.13 — Accelerometer Sensor
Section 2.14 — Backlight Control Interface
Section 2.15 — IrDA
Section 2.16 — Demo List
Section 2.17 — Device File List

## 2.1 Dependence

### 2.1.1 Software Resources

Linux system of Sitsang / PXA255 evaluation board includes following images:

| *Flash Bank Capability* | *File Name* | *Description* |
|---|---|---|
| Dual-Bank | redboot.bin | Boot loader |
| | sitsang_fs.jffs2 | Root file system for Linux |
| | zImage | Linux kernel |
| Single-Bank | redboot_single.bin | Boot loader |
| | sitsang_fs_single.jffs2 | Root file system for Linux |
| | zImage_single | Linux kernel |

**Table 1: Linux Binary Packages List**

**NOTE 1**: User can find all the files in the BSP disc for Sitsang board.
**NOTE 2:** This version Redboot extends the available flash space size for user. It enables the selection between using two flash banks and using only one flash bank. There are two Redboot binary images in this distribution. One is redboot.bin, which uses both flash banks on Sitsang board; and the other is redboot_single.bin, which uses only one flash bank on Sitsang. The single bank version Linux makes it possible that two different operating systems can be burned in Sitsang board's two banks respectively. As a result, there are also two Linux kernel binary images correspond to the two Redboot binary images, which are named "zImage" and "zImage_single". Refer to Section 2.12 for details. The installation process of both dual-bank and single-bank is almost identical, except some small adjustment of command line parameters.

### 2.1.2 Hardware Resources

In order to using/debugging the Linux system of Sitsang / PXA255 evaluation board, below hardware components are required:

| *Hardware* |
|---|
| Power Supplier |
| Bi-direction 9-pin COM port cable for BTUART |
| Bi-direction JTAG cable for FFUART |
| Cross over Ethernet cable |
| Desktop PC running Windows or Redhat(Mandrake) Linux 7.2 (named **host**) |
| Sitsang-PXA255 Evaluation Board (named **Sitsang board**) |

**Table 2: Hardware List**

## 2.2 Cable Connection

a) Connect power supplier to the Sitsang board, and ensure power is provided.
b) Connect JTAG cable between host and Sitsang board. Refer to **Sitsang / PXA255 Evaluation Platform User's Guide** for JTAG cable connection.
c) Connect one end of 9-pin cable to J21 (JTAG/FFUART Port) and the other end to the host's serial port.
d) Connect Ethernet cross cable between host and the Sitsang board.

## 2.3 Linux Loading Steps

This section describes how to load Linux into Sitsang board. The example is for dual-bank scheme, and the instructions for single-bank scheme can be found in NOTE section.

Step 1. Burn *redboot.bin* into flash starting at 0x0000_0000 from host to Sitsang board. User may use the Jflash utility for Sitsang. For how to use Jflash for Windows, please refer to **Sitsang / PXA255 Evaluation Platform User's Guide**.
User can use Jflash for Linux to burn the 'redboot.bin' into Sitsang Board Flash Bank 0 by inputting the following commands:

- Copy the Jflashmm-linux to current directory.
- %***chmod 755 ./Jflashmm-linux***
- %***./Jflashmm-linux sitsang redboot.bin***

NOTE: If you install single-bank RedBoot into board, replace *redboot.bin* with *redboot_single.bin*.

Step 2. Start user's favorite terminal emulator on host (such as HyperTerminal in Windows, or minicom in Linux). The default settings should be:

| *Item* | *Setting* |
|---|---|
| Bit Per Second (bps) | 115200 |
| Data Bits | 8 |
| Parity | None |
| Stop bit | 1 |
| Flow Control (software & hardware) | None |
| Emulation | VT100J or VT102 |

**Table 3: Settings of the HyperTerminal or minicom**

Reset Sitsang board, the following information is displayed on terminal emulator:

```
+
ARM eCos
RedBoot(tm) debug environment - built 04:52:36, Dec  4 2002
Platform: Intel(R) Sitsang Development Platform RedBoot* Debug Monitor ()
Copyright (C) 2000, Red Hat, Inc.
```

*RAM: 0xa0000000-0xa4000000*
*FLASH: 0x00100000 - 0x04000000, 128 blocks of 0x00040000 bytes each.*
*FLASH configuration checksum error or invalid key*
*LAN91C96: The current MAC address is 00 5B 38 BF 33 40*
*Can't get BOOTP info – network disabled!*
*RedBoot>*

For the first time of running RedBoot on Sitsang board after burning, there might be an error message: "*FLASH configuration checksum error or invalid key*". It will disappear after Redboot configuration. RedBoot configuration help can be got from section 2.5.2.

Step 3. Initialize the flash on Sitsang board. Enter commands in the terminal emulator as follows:

*RedBoot>* ***fis init***
*About to initialize [format] FLASH image system - are you sure (y/n)?* ***y***
**** Initialize FLASH Image System*
*Warning: device contents not erased, some blocks may not be usable*
*... Unlock from 0x03fc0000-0x04000000: block: 3fc0000,block_size: 40000,blocks: .*
*... Erase from 0x03fc0000-0x04000000:*
*addr:3fc0000,data:a3fb0000,size:300 .*
*... Program from 0xa3fb0000-0xa3fb0400 at 0x03fc0000: .*
*... Lock from 0x03fc0000-0x04000000: .*

Then user can use '*fis list*' command to see flash partitions list. Enter the following commands in the terminal emulator:

*RedBoot>* ***fis list***

| *Name* | *FLASH addr* | *Mem addr* | *Length* | *Entry point* |
|---|---|---|---|---|
| *RedBoot* | *0x00000000* | *0x00100000* | *0x00040000* | *0x00000000* |
| *RedBoot config* | *0x03F80000* | *0x03F80000* | *0x00040000* | *0x00000000* |
| *FIS directory* | *0x03FC0000* | *0x03FC0000* | *0x00040000* | *0x00000000* |

**NOTE 1**: Since the Linux kernel binary image shipped with this release is configured with kernel command string "root=/dev/mtdblock3", it may fail to load root fs if the number of entries in the Redboot flash file system does not match the list above.
**NOTE 2**: If you installed single-bank RedBoot, the message is different. Refer 2.12 for help.

Step 4. Load the file system image *sitsang_fs.jffs2* to Sitsang board's SDRAM from host:

*Redboot>* ***load -r -m xmodem -b 0xa0800000***

Then initiate the file (zImage) transfer using 'xmodem' as the transfer protocol. Please work **quickly** since the Redboot has a timeout value for loading xmodem data..

a) For HyperTerminal:
1. Click **Transfer** -> **send file...**
2. Select Xmodem in **Protocol** list, and click **Browse** button to select the *sitsang_fs.jffs2* file.
3. Click **Send** button to start transferring.

b) For minicom:
1. Press key: **Ctrl-A S**, and use **direction key** to select xmodem, then press **ENTER**.
2. Select the sitsang_fs.jffs2 file to upload. (User can use **direction key** to move, press the space key **twice** to enter the directory, press the space key **once** to select the file, finally user can press enter to confirm the selection and start the transfer.)

OPTIONALLY, if TFTP has been started on host (Refer section 2.5 for help), try the following command:

*Redboot>* ***load -r -m TFTP -b 0xa0800000 sitsang_fs.jffs2***
*Raw.file loaded 0xa0800000-0xa3000000*

**NOTE 1:** For how to setup a TFTP server on Linux, please refer to section 2.5**.**

**NOTE 2:** Sometimes, an error may occur when the first time that Sitsang board tries to access LAN in active mode, such as to ping other computer from Sitsang board. It only happens on some special hosts.
**Workaround:** *Ping* the Sitsang board from host, or use HUB/SWITCH to connect host and Sitsang board.

**NOTE 3:** After Step 4, the image is in Sitsang board's SDRAM. If user powers off the Sitsang board, all contents in SDRAM will loss. Please finish step 5 before powering off.

**NOTE 4:** It takes about 2 hours to transfer the sitsang_fs.jffs2 with xmodem protocol. Please make sure the charger is plugged during transfer. And it takes about 5 minutes to transfer with TFTP protocol. Please drink a cup of tea!

**NOTE 5:** If you employ single-bank scheme, replace the file "*sitsang_fs.jffs2*" with "*sitsang_fs_single.jffs2*"**.**

**NOTE 6:** If you reprogram a board with dual-bank scheme after programming as single-bank, you should unlock flash manually using this command:

*Redboot>* ***fis unlock -f 0x00200000 -l 0x02800000***

Step 5. Burn the downloaded file system image into flash from SDRAM and verify flash partition:

*Redboot>* ***fis create JFFS2 -b 0xa0800000 -l 0x02800000 -f 0x00200000***

*... Erase from 0x00200000-0x02a00000: .....*
*... Program from 0xa0800000-0xa3000000 at 0x00200000: .....*
*addr:200000,data:a0800000,size:40000.*
*.....*
*...Unlock from 0x03fc0000-0x04000000: block: 3fc0000,block_size:*
*40000,blocks: .....*
*... Erase from 0x03fc0000-0x04000000: .....*
*... Program from 0xa3fb0000-0xa3ff0000 at 0x03fc0000: .....*
*addr:3fc0000,data:a3fb0000,size:40000*
*... Lock from 0x03fc0000-0x04000000:*

*Redboot>* ***fis list***

| *Name* | *FLASH addr* | *Mem addr* | *Length* | *Entry point* |
|---|---|---|---|---|
| *RedBoot* | *0x00000000* | *0x00100000* | *0x00040000* | *0x00000000* |
| *RedBoot config* | *0x03F80000* | *0x03F80000* | *0x00040000* | *0x00000000* |
| *FIS directory* | *0x03FC0000* | *0x03FC0000* | *0x00040000* | *0x00000000* |
| *JFFS2* | *0x00200000* | *0x00200000* | *0x02800000* | *0x00000000* |

**NOTE 1:** It takes about ten minutes to complete this step.
**NOTE 2**: If you employ single-bank scheme, the parameters is a little different. Use this command:
*Redboot>* ***fis create JFFS2 -b 0xa0800000 -l 0x01c00000 -f 0x00200000***

Step 6. Type the following command to download the kernel image to Sitsang board's SDRAM from host:

*Redboot>* ***load -r -m xmodem -b 0xa0200000***

Instruct user's terminal emulator to initiate the file (zImage) transfer using 'xmodem' protocol.

Refer to Step 4 for details.

OPTIONALY, if user's environment is configured for TFTP download, try the following command:

*Redboot>* ***load -r -m TFTP -b 0xa0200000 zImage***
*Raw file loaded 0xa0200000- 0xa02c761c*
**NOTE**: If you employ single-bank scheme, only replace zImage with zImage_single.

Step 7. After the kernel image has been successful transferred from host to Sitsang board's SDRAM, burn the downloaded kernel image onto flash and verify flash partition:

*Redboot>* ***fis create linux -b 0xa0200000 -l 0x00100000 -f 0x00100000 -r 0xa0200000***

*... Erase from 0x00100000-0x00200000: .....*
*... Program from 0xa0200000-0xa0300000 at 0x00100000:*
*addr:100000,data:a0200000,size:40000*
*.. .....*
*... Unlock from 0x03fc0000-0x04000000: block: 3fc0000,block_size:*
*40000,blocks: .....*
*... Erase from 0x03fc0000-0x04000000: .....*
*... Program from 0xa3fb0000-0xa3ff0000 at 0x03fc0000:*
*addr:3fc0000,data:a3fb0000,size:40000*
*... Lock from 0x03fc0000-0x04000000: .....*

*RedBoot>* ***fis list***

| *Name* | *FLASH addr* | *Mem addr* | *Length* | *Entry point* |
|---|---|---|---|---|
| *RedBoot* | *0x00000000* | *0x00100000* | *0x00040000* | *0x00000000* |
| *RedBoot config* | *0x03F80000* | *0x03F80000* | *0x00040000* | *0x00000000* |
| *FIS directory* | *0x03FC0000* | *0x03FC0000* | *0x00040000* | *0x00000000* |
| *JFFS2* | *0x00200000* | *0x00200000* | *0x02800000* | *0x00000000* |
| *linux* | *0x00100000* | *0xA0200000* | *0x00100000* | *0x00000000* |

**NOTE**: If you employ single-bank scheme, the burn commands is same as dual-bank scheme. The output of “fis list” is different.

Step 8. Every time board RESET or power up, start up the kernel with the following Redboot commands:

*Redboot>* ***fis load linux***
*Redboot>* ***go 0xa0200000***

**Tip:** User can save these commands in flash as auto-run scripts to avoid typing them every time. Please refer to section 2.5.2 for help.

User should see the following messages on the terminal emulator:

*Uncompressing Linux.....*
*< kernel displays a lot of debug messages >*
*.....*
*Linux login:* ***root***
*[root@Linux /root]$*

**NOTE**: It may take longer time for the first time the kernel boot up, because some unused flash blocks should be erased at that time. Subsequent kernel boot should be quicker.

When Linux is running, Led D28 acts as the heartbeat indicator of system and D34 indicates the status that system is busy.

Step 9. Shutdown or reboot the Linux System:
a) To shutdown:
*[root@Linux /root]$* ***shutdown –r now***
After Redboot startups again, user can power off the Sitsang board now.

b) To reboot:
*[root@Linux /root]$* ***reboot***

**NOTE:** Redboot sets up the CPU core frequency based on the value of S7 in JSSR. Refer to **Sitsang / PXA255 Evaluation Platform User's Guide** for more information.

| *S7 Value* | *0~3* | *4~7* | *8~11* | *12-15 (PXA255 only)* |
|---|---|---|---|---|
| Turbo Mode Clock (MHz) | 300 | 400 | 200 | 400 |
| Run Mode Clock (MHz) | 200 | 200 | 200 | 400 |
| SDRAM clock (MHz) | 100 | 100 | 100 | 100 |

**Table 4: Frequency configuration based on the value of S7**

## 2.4 Network Configuration

User may configure the network by modifying the following files in Sitsang board's Linux system based on user's own requirement:

/etc/sysconfig/network ----network options
/etc/sysconfig/network-scripts/ifcfg-eth0 ----netcard eth0 options

Normally, only IPADDR, NETMASK, BROADCAST, GATEWAY and BOOTPROTO need to be changed.
Here:
IPADDR is user's IP address; NETMASK is user's network mask; BROADCAST is user's broadcast address; GATEWAY is user's gateway address and BOOTPROTO is the dynamic IP allocation protocol. Please refer to user's own network administrator for more details.

**TIP 1**: The text editor named '**edit**' or '**vi**' can be used to edit these files. '**edit**' is more easy to use for new Linux users**.** For how to use '**edit**' or '**vi**', please refer to the help document of them. User can input the following command to edit the file.

% ***edit /etc/sysconfig/network***
or
% ***vi /etc/sysconfig/network***

After user configuring the network, restart the network service by entering:

% ***/etc/rc.d/init.d/inet stop***
% ***/etc/rc.d/init.d/inet start***

OPTIONALLY, user can add user's DNS entries in

/etc/resolv.conf

**TIP 2:** DHCP is supported for dynamic IP allocation now. If the BOOTPROTO is dhcp, the IPADDR, NETMASK, BROADCAST, and GATEWAY won't be used because DHCP client can get this information from DHCP server.

## 2.5 Configuring User's Development Environment for TFTP Downloading

### 2.5.1 Configuring TFTP server for RedHat (Mandrake) Linux on host:

TFTP server is used for image downloading to Sitsang board. The recommend version is 0.29 or higher.
**The bound TFTP server version with Mandrake 8.0 may cause transfer error.**
To enable TFTP on RedHat v6.1:

Step 1. Uncomment the following line in the file /etc/inetd.conf

```
tftp    dgram   udp     wait    root    /usr/sbin/tcpd  in.tftpd
```

Step 2. Create the TFTP default directory:

```
% mkdir /tftpboot
```

Step 3. Copy the necessary files to /tftpboot/:

```
% cp sitsang_fs.jffs2 zImage   /tftpboot/
```

Step 4. Restart inetd server:

```
% killall –HUP inetd
```

To enable TFTP on RedHat v7.2 or Mandrake v7.2, or higher versions:

Step 1. Edit '/etc/xinetd.d/tftp', and modify the following lines:

```
service tftp
{
# enable , yes => no
disable = no
socket_type = dgram
...
# set root directory for tftp user, here is /tftpboot/
 server_args = -s /tftpboot/
}
```

Step 2. Copy the necessary files to /tftpboot/

```
% cp sitsang_fs.jffs2 zImage /tftpboot/
```

Step 3. Restart xinetd server:

```
%/etc/rc.d/init.d/xinetd   restart
```

### 2.5.2 Configuring Redboot on Sitsang board:

User can configure the network and auto-run script for Redboot with *fconfig* command.

#### *2.5.2.1* Configuring the network for Redboot

Step 1.
```
Redboot> fconfig
 Run script at boot: false false
  Use BOOTP for network configuration:false  false
 Local IP address: 0.0.0.0  192.168.1.100
 Default server IP address: 0.0.0.0  192.168.1.1
 GDB connection port: 0 9000
```

```
Network debug at boot time: false  false
Update Redboot non-volatile configuration – are you sure (y/n)?  y
...Unlock from 0x03f80000-0x03fc0000: .
...Erase from 0x03f80000-0x03fc0000: .
...Program from 0xa000bd78-0xa000c178 at 0x03f80000: .
...Lock from 0x03f80000-0x03fc0000: .
```

**NOTE:** Here 192.168.1.100 is the IP address of Sitsang board and 192.168.1.1 is the IP address of host. User can modify them based on user's own requirement.

Since there may be a duplicate assignment with pre-configured MAC address of Sitsang board in user's own network, user can set a unique MAC address for the Ethernet card:

```
Redboot> set mac address <6 byte hex value - i.e. 004e21234567>
```

Step 2. Reboot the Sitsang board and reset the Redboot:

```
+
ARM eCos
RedBoot(tm) debug environment - built 04:52:36, Dec  4 2002
Platform: Intel(R) Sitsang Development Platform RedBoot* Debug Monitor ()
Copyright (C) 2000, Red Hat, Inc.

RAM: 0xa0000000-0xa4000000
FLASH: 0x00100000 - 0x04000000, 128 blocks of 0x00040000 bytes each.
LAN91C96: The current MAC address is 00 5B 38 BF 33 40
IP: 192.168.1.100, Default server: 192.168.1.1
RedBoot>


Redboot> reset
+
ARM eCos

RedBoot(tm) debug environment - built 04:52:36, Dec  4 2002
Platform: Intel(R) Sitsang Development Platform RedBoot* Debug Monitor ()
Copyright (C) 2000, Red Hat, Inc.

RAM: 0xa0000000-0xa4000000
FLASH: 0x00100000 - 0x04000000, 128 blocks of 0x00040000 bytes each.
LAN91C96: The current MAC address is 00 5B 38 BF 33 40
IP: 192.168.1.100, Default server: 192.168.1.1
```

### 2.5.2.2 Configuring the boot script for Redboot

User can set the boot script to auto run some commands when Redboot starts up:

```
Redboot> fconfig
Run script at boot: false true
Boot script:
Enter script, terminate with empty line
>>fis load linux
>>go 0xa0200000
>>
Boot script timeout (1000ms resolution): 0 5
Use BOOTP for network configuration:false  false
Local IP address: 0.0.0.0  192.168.1.100
Default server IP address: 0.0.0.0  192.168.1.1
GDB connection port: 0 9000
Network debug at boot time: false  false
Update Redboot non-volatile configuration – are you sure (y/n)?  y
...Unlock from 0x03f80000-0x03fc0000: .
...Erase from 0x03f80000-0x03fc0000: .
```

*...Program from 0xa000bd78-0xa000c178 at 0x03f80000: .*
*...Lock from 0x03f80000-0x03fc0000: .*

## 2.6 Mounting NFS Partition and RAMFS Partition

The Network File System (NFS) was developed to allow machines to mount a disk partition on a remote machine as if it were on a local hard drive. This allows for fast, seamless sharing of files across a network.

A RamDisk is a portion of memory that can be allocated to use as a partition. Or, in other words, it takes some of user's memory and can be treated as a hard drive that user can format, mount, save files etc. The RAMFS is the file system for RamDisk.

### 2.6.1 Mounting NFS partition

Detailed information for setting up user's NFS server and client can be obtained at http://www.linuxdoc.org/HOWTO/NFS-HOWTO/.

#### *2.6.1.1* Setting up an NFS server

Step 1. Edit the file /etc/exports to share a volume:

/home/fileserver <enter Sitsang board's ip address here>(rw,no_root_squash)

Step 2. Initialize NFS services (RedHat 6.1 and above):

% ***/etc/rc.d/init.d/nfs stop***
% ***/etc/rc.d/init.d/nfs start***

**NOTE: U**ser may optionally follow the instructions in the NFS-HOWTO.

Step 3. After modifying the /etc/exports file, execute the following command:

% ***exportfs -rav***

#### *2.6.1.2* Setting up an NFS client

% ***mount -o nolock <enter NFS server's ip address here>:/home/fileserver /mnt***

### 2.6.2 Mounting RAMFS partition

System mounts a ramfs file system on /tmp when booting by default. User can mount another ramfs file system on other mount point as following:

% ***mkdir -p /ram1***
% ***mount -t ramfs none /ram1***

## 2.7 Loading QTOPIA Application

Step 1. If it is the first time to run the QTOPIA application, user may set the touch screen calibration values:

%***/etc/rc.d/init.d/qtopia stop***
% ***/usr/qpe/bin/ts_calibrate***
*xres = 640, yres = 480*
*Top left : X = 3490 Y = 621*
*Top right: X = 3482 Y = 3490*
*Bot right: X = 698 Y = 3411*
*Bot left : X = 709 Y = 613*

*Middle: X = 2063 Y = 1998*
*-59.825392 -0.003002 0.190514*
*526.058573 -0.136547 -0.000442*
*Calibration constants: -3920716 -196 12485 34475774 -8948 -28 65536*
**NOTE:** Please verify the values of X and Y for each point. The X value of two top points or two bottom points should be almost same; The Y value of two left points or two right points should be almost same; The X and Y of Middle point should be about 2000.
***%/etc/rc.d/init.d/qtopia start***

Or, go to Step 2 directly.

Step 2. Run and stop the QTOPIA demo application manually:

- Run:
% ***/etc/rc.d/init.d/qtopia start***

- Stop:
% ***/etc/rc.d/init.d/qtopia stop***

*[This way to stop QTOPIA is just for the sake of debugging. Sometimes, the system can't be rebooted successfully after doing this command. The right way to stop or reboot QTOPIA is explained in NOTE 1]*

**NOTE 1**: If the QTOPIA is running, user should use the Setting->Shutdown application in QTOPIA to stop QTOPIA or reboot it.

**NOTE 2**: If user wants to write user's own QTOPIA application, please refer to: http://qpe.sourceforge.net/ or http://www.trolltech.com/.

## 2.8 USB Client

Most of the USB client drivers are compiled as Linux kernel module, so user **needs** to update the modules when updating the Linux kernel. The following is the module list required by USB Client.

| *Kernel Module Name* | *Description* | *Location in Sitsang File System* |
|---|---|---|
| usbdcore.o | Core support for USB device | /lib/modules/2.4.19-rmk4-pxa2-alvin2/kernel/drivers/usb/device/usbdcore.o |
| serial_fd.o | USB device serial function support | /lib/modules/2.4.19-rmk4-pxa2-alvin2/kernel/drivers/usb/device/serial_fd/serial_fd.o |
| pxa_bi.o | USB Device bus interface support for PXA255 | /lib/modules/2.4.19-rmk4-pxa2-alvin2/kernel/drivers/usb/device/bi/pxa_bi.o |
| bsd_comp.o | PPP BSD-Compression | /lib/modules/2.4.19-rmk4-pxa2-alvin2/kernel/drivers/net/bsd_comp.o |
| ppp_async.o | PPP support for async serial ports | /lib/modules/2.4.19-rmk4-pxa2-alvin2/kernel/drivers/net/ppp_async.o |
| ppp_deflate.o | PPP Deflate compression | /lib/modules/2.4.19-rmk4-pxa2-alvin2/kernel/drivers/net/ppp_deflate.o |

**Table 5: Module List for USB Client**

For compatible reason, only Linux host PC can communicate with Sitsang board through USB client. The Linux distribution running on Linux host PC is Mandrake 8.0.

### 2.8.1 Communicating with *Cat*

- Host PC:
  If the Linux distribution is **Mandrake**, user need do the following command to ensure USB service is running. For Redhat, it is unnecessary.

%***/etc/rc.d/init.d/usb restart***

Then load the usbserial module. This step is required for all Linux distributions.
%***modprobe usbserial vendor=0x8086 product=0xA6CD***

- Sitsang:
  % ***/bin/usbclient start***

- Cat with ttyUSB0
  - Re-plug the USB cable now, the USB client led will light.
  - Host PC side:
    %***cat /dev/ttyUSB0***
  - Sitsang side:
    %***cat /etc/passwd >/dev/ttyUSB0***

  Or,
  - Replug the USB cable.
  - Sitsang side:
    %***cat /dev/ttyUSB0***
  - Host PC side:
    %***cat /etc/passwd>/dev/ttyUSB0***

**NOTE:** If user uses the other Sitsang Board as USB Host, the USB device file for USB host is /dev/ttyUSB1 instead of /dev/ttyUSB0.

### 2.8.2 Communicating with PPP

Using the usbserial driver, user can setup ppp server.

- Host PC side:
  %***pppd /dev/ttyUSB0 noauth 192.168.1.1:192.168.1.100***
  **NOTE:** Here, 192.168.1.1 is the IP address for Host PC; 192.168.1.100 is the IP address for Sitsang board.

- Sitsang side:
  %***pppd /dev/ttyUSB0 defaultroute usepeerdns***

**Warning:** User must finish the above commands quickly to avoid *pppd*'s timeout.

Now, user can ping Sitsang board from Host PC, or ping Host PC from Sitsang via ppp connection on USB.

**NOTE**: If user uses one Sitsang board as Host and another Sitsang board as Client. The USB Host device is /dev/ttyUSB1 instead of /dev/ttyUSB0 for Sitsang Host.

## 2.9 USB Host

Only USB keyboard and mouse drivers are compiled in the Linux kernel by default. If user wants to support more USB device, please recompile the kernel and select the kernel options for customized device. Please refer to Section 3 for building Linux Kernel help.

Two Sitsang boards can be connected with USB Cable and communicate with serial protocol. And the usbserial.o module is required for USB host.

| *Kernel Module Name* | *Description* | *Location in Sitsang File System* |
|---|---|---|
| usbserial.o | USB serial converter support | /lib/modules/2.4.19-rmk4-pxa2-alvin2/kernel/drivers/usb/serial/usbserial.o |

**Table 6: Module List for USB Host**

To load the module:

%***modprobe usbserial  vendor=0x8086 product=0xa6cd***

**NOTE 1**: If user uses USB keyboard as input device while QTOPIA is running, user must do the following commands to input correct characters on the console.

%***/etc/rc.d/ini.td/qtopia stop***

%***ts_test*** (Use CTRL-C to interrupt.)

Then USB keyboard can be used for inputting.

**NOTE 2:** In this distribution, QTOPIA doesn't support USB input device, such as keyboard and mouse.

## 2.10 AC97

User can use some GUI applications in QTOPIA to play divx video stream or mp3 audio file. Moreover, some audio-related applications based on text mode are supplied in this distribution. User can use these applications to play raw wav file, record and tune the volume.

### 2.10.1 Record via microphone

- Plug the LINE-OUT and MIC-IN cables into the connectors of the Sitsang board.
- %***cat /dev/dsp > /tmp/a.wav***
- Say something by microphone, and use "CTRL-C" to terminate it.

The recorded sound is saved as a raw wave file "a.wav" in the /tmp directory. User can play this file by the typing the following command.

- %***cat /tmp/a.wav >/dev/dsp***

### 2.10.2 Tune the volume

- Query the volume of AC97 codec

  %***mixer [option]***

  Here, option can be one of the items: ***vol, pcm, speaker, line, mic*** and ***igain***. For example, user can use the following command to query the current volume of mic.

  %***mixer mic***

- Tune the volume of AC97 codec

  %***mixer [option] [value]***

  Here, option may be one of these items: ***vol, pcm, speaker, line, mic*** and ***igain***; value is the setting of volume, which ranges from 0 to 100. For mic, please refer to the following table.

| *mic volume* | *10db booster* | *20db booster* | *mute* |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 1 --- 100 | 0 | 0 | 0 |
| 101 --- 200 | 1 | 0 | 0 |
| 201 --- 300 | 0 | 1 | 0 |
| 301 --- 400 | 1 | 1 | 0 |

**Table 7: Mic Volume Setting List**

For example, user can use the following command to set the volume of dual channels of line-in to 60/100.

%***mixer line 60***

Or, set the volume of left channel to 10 and right channel to 80

%***mixer line  10:80***

For mic, because it is a mono source, there is only one volume channel.

**NOTE**: The mute bit of any volume will be set only if the [value] option is 0 or 0:0. Otherwise, for example, the [value] option is 0:80 will only set the left channel volume to 0. For the volumes of record sources, such as ***mic***, ***line***, and ***pcm***, 0 means gain level is +12DB, while as to the volumes of ***vol***, ***speaker*** and ***igain***, 0 means gain level is 0DB.  Please refer to the CS4201 data sheet for more details.

### 2.10.3 Set the record source

- Query the current record source
  %***mixer ?rec***
- Add a record source
  %***mixer +rec [option]***
  Here, option can be one of the following items***: vol, pcm, speaker, line, mic*** and ***igain***. For example, user can use the following command to select mic as the record source.
  %***mixer +rec mic***

**NOTE**: CS4201 supports only one record source at any time. There are two record sources used on Sitsang: ***mic*** and ***line***.

### 2.10.4 Change the A/D and D/A frequency

There are some A/D and D/A in the audio codec ---- CS4201. The sample frequency can be set by using "*speed*" tool. The valid sample frequency is one of the following: 8000, 11025, 16000, 22050, 32000, 44100, and 48000. Any other value in the range of 8000 to 48000 will be adjusted to the closest one of the 7 values mentioned above. For example,

%***speed 22000***

The A/D and D/A frequency will be set to 22050 Hz.

## 2.11 CF Card

There are two kinds of CF Cards are built as kernel modules by default: NE2000 compatible CF Network Adapter and Spectrum24® 802.11 Wireless Networker. These modules **should** be updated every time when kernel is recompiled. If user wants to support more kinds of CF Cards, please enable the kernel options for customized device and recompile kernel to enable it. And then add the related entry in /etc/pcmcia/config.opts in Sitsang board file-system. The following is the module list required by these two default devices.

| *Kernel Module Name* | *Description* | *Location in Sitsang File System* |
|---|---|---|
| 8390.o | NE2000 generic support | /lib/modules/2.4.19-rmk4-pxa2-alvin2/kernel/drivers/net/8390.o |
| pcnet_cs.o | NE2000 compatible PCMCIA support | /lib/modules/2.4.19-rmk4-pxa2-alvin2/pcmcia/pcnet_cs.o |
| hermes.o | Generic support for this wireless card. | /lib/modules/2.4.19-rmk4-pxa2-alvin2/pcmcia/hermes.o |
| orinoco.o | Driver framework support. | /lib/modules/2.4.19-rmk4-pxa2-alvin2/orinoco.o |
| spectrum_cs.o | Spectrum24® CF Card support. | /lib/modules/2.4.19-rmk4-pxa2-alvin2/spectrum_cs.o |

**Table 8: Module List for CF Card**

To start CF card service:

%***/etc/rc.d/init.d/pcmcia start***

To stop CF card service:

%***/etc/rc.d/init.d/pcmcia stop***

User can modify the pre-assigned netcard options in "/etc/pcmcia/network.opts", such as IPADDR, NETMASK and NETWORK.

## 2.12 Redboot Flash Partition Table

There are two flash banks on Sitsang Board, which enable two different schemes mentioned at Section 2.1.1. For these two schemes, there are two kinds of flash partition tables correspondingly. One is for

dual-bank scheme; the other is for single-bank scheme. Refer **Sitsang / PXA255 Evaluation Platform User's Guide** for more details about system address map.

### 2.12.1 Dual Flash Banks Scheme

| *Flash Bank* | *Begin* | *End* | *Content* |
|---|---|---|---|
| Bank 0 | 0x00000000 | 0x00040000 | Redboot |
| | 0x00040000 | 0x00100000 | Free Space |
| | 0x00100000 | 0x00200000 | Linux Kernel Image |
| | 0x00200000 | 0x02000000 | File System |
| Bank 1 | 0x02000000 | 0x03F80000 | Free Space |
| | 0x03F80000 | 0x03FC0000 | Redboot Config |
| | 0x03FC0000 | 0x04000000 | FIS Directory |

**Table 9: Two Flash Banks Scheme Partition Table of Redboot**

### 2.12.2 Single Flash Bank Scheme

| *Flash Bank* | *Begin* | *End* | *Content* |
|---|---|---|---|
| Bank 0 | 0x00000000 | 0x00040000 | Redboot |
| | 0x00040000 | 0x00100000 | Free Space |
| | 0x00100000 | 0x00200000 | Linux Kernel Image |
| | 0x00200000 | 0x01F00000 | File System |
| | 0x01F00000 | 0x01F80000 | Free Space |
| | 0x01F80000 | 0x01FC0000 | Redboot Config |
| | 0x01FC0000 | 0x02000000 | FIS Directory |
| Bank 1 | 0x02000000 | 0x04000000 | Can't be used |

**Table 10: Single Flash Bank Partition Table of Redboot**

## 2.13 Accelerometer Sensor

The device file for Accelerometer Sensor driver is "/dev/acce". User can open this device file and read the data from it. Refer to the source code in Linux Kernel "./drivers/misc/accelerometer.c" for programming help.

User can make the device file manually by the following command.

```
%mknod    /dev/acce      c     10     241
```

## 2.14 Backlight Control Interface

The brightness of LCD's backlight can be controlled by the generic LCD control interface. The device file for backlight control is "*/dev/lcdctrl*", whose major number is dynamic assigned and the default value is 254. User can open the device file then adjust the brightness by system call "*ioctl*". The *ioctl* command numbers are defined in the head file "*video/lcdctl.h*" (kernel version 2.4.19 or highe).

For example, the following code will set the brightness to 50% of maximal brightness.

```
unsigned int value = 50;  //the value is the percent of maximal brightness, ranging from 0 to 100.
int fd = open("/dev/lcdctrl", O_WRONLY);
ioctl( fd, _LCDCTRL_IOCTL_BRIGHTNESS, value);
```

## 2.15 IRDA

Sitsang supports both SIR and FIR. IrDA is a network device. User may use *ifconfig* to control the IrDA device.
For example, to turn on the IrDA device:

```
% ifconfig irda0 up
```

To turn off the IrDA device, use:

```
% ifconfig irda0 down
```

There is an application, **irxfer**, to transfer files. The other side may be another Sitsang board or an IrDA enabled general purpose PC.
For example, to send a file named test from Sitsang A to Sitsang B

Sitsang B:

```
% irxfer
```

Sitsang A:

```
% irxfer testfile
```

The file "testfile" will be saved in the directory "/tmp" on Sitsang B.
It is same to transfer files between Sitsang and general purpose PC via IrDA connection.

## 2.16 Demo List

There are some demo applications in QTOPIA shipped with this release.

| *Application Icon in QTOPIA* | *Description* |
|---|---|
| Ball | A game of Accelerometer Sensor Demo. |
| Dr. Z video Player | MPEG-4 DIVX video player. |
| MPEG Player | Play a mp3 or MPEG-1 file. |
| Web Browser | A Web browser - Konqueror/Embedded. |
| XMMS | A winamp-like player, support mp3, MPEG-1 files, ShoutCAST audio stream media. |

**Table 11: Demo List**

## 2.17 Device File List

The following table lists all device files for all kinds of drivers supported in the distribution.

| *File name* | *Device Type* | *Major Number* | *Minor Number* | *Description* |
|---|---|---|---|---|
| /dev/acce | Char | 10 | 241 | Accelerometer Sensor |
| /dev/input/event0 | Char | 13 | 64 | Joystick & Softbutton |
| /dev/input/event1 | Char | 13 | 65 | ADS7846 TouchScreen |
| /dev/ttyUSB0 | Char | 188 | 0 | USB Client Serial Driver |
| /dev/ttyUSB1 | Char | 188 | 1 | USB Host Serial Device 1 |
| /dev/ttyUSB2 | Char | 188 | 2 | USB Host Serial Device 2 |
| /dev/ttyS0 | Char | 4 | 64 | FFUART |
| /dev/ttyS1 | Char | 4 | 66 | BTUART |
| /dev/dsp0 | Char | 14 | 3 | AC97 OSS Driver |
| /dev/mtdblock3 | Block | 31 | 3 | File System Partition |
| /dev/lcdctrl | Char | 254 (Dynamic) | 0 | LCD Backlight Control |

**Table 12: Device File List**

# 3 Linux for Sitsang from Scratch

This chapter describes the development of Linux operating system for the Sitsang / PXA255 Evaluation platform.

A host pc running RedHat Linux 7.2 or Mandrake Linux 7.2 is necessary; and a sound knowledge of the Linux operating system and C programming language is required for using this document effectively.

This chapter contains:

Section 3.1 — Dependence
Section 3.2 — Installing the tool chain
Section 3.3 — Building Redboot from source code
Section 3.4 — Building Linux kernel from source code
Section 3.5 — Creating a JFFS2 file system image

## 3.1 Dependence

| | *File Name* | *Description* | |
|---|---|---|---|
| **Tool Chain** | xscale-arm-linux-toolchain.tgz | Tools for cross-compile | |
| | gnupro-bin-glibc2.2.4.tgz | Binary package for gnupro tool | |
| | mkfs.jffs2 | JFFS2 make file system utility | |
| **Source Code** | RedBoot_Intel.tgz | Source code of Redboot porting on sitsang | |
| | linux-2.4.19.tar.bz2 | Linux source code from www.kernel.org | |
| | patch-2.4.19-sitsang2.gz | The necessary patchs for Sitsang board includes the following three patches | |
| | | patch-2.4.19-rmk4.gz | patches by Russell King |
| | | diff-2.4.19-rmk4-pxa2.gz | patches by Nicolas Pitre |
| | | diff-2.4.19-rmk4-pxa2-alvin2 | patches by Alvin Tang |
| | sitsang_fs.tgz | Root file system source code | |

**Table 13: Linux Tool Chain and Source Codes List**

**NOTE 1**: This document assumes that user copy all source code packages to the directory **/tmp**, and user's install target directory is **/home/sitsang**. Please change them to fit user's own system.
Create the install target directory:

```
%mkdir –p  /home/sitsang
```

**NOTE 2**: Here, '%' is the command prompt of Linux operating system. User can find all the files in the BSP disc for Sitsang board.

**NOTE 3**: Root privilege is **required** to do the following steps.

## 3.2 Installing the Tool Chain

Step 1. Untar and install the xscale-arm-linux-toolchain:

```
% cd /
% tar zxvf  xscale-arm-linux-toolchain.tgz
```

After installation, the arm-linux cross-compile tool is in the directory: /usr/local/arm-linux/bin.

Step 2. Add user's relative path to user's PATH setting.

For bash user:
% ***echo "export PATH=/usr/local/arm-linux/bin:${PATH}" >>~/.bashrc***
% ***source ~/.bashrc***

For csh user:
% ***echo "setenv PATH /usr/local/arm-linux/bin:${PATH}">>~/.cshrc***
% ***setenv PATH /usr/local/arm-linux/bin:${PATH}***

Step 3. Untar the gnupro binary package:
% ***cd /usr***
% ***tar zxvf /tmp/gnupro-bin-glibc2.2.4.tgz***

## 3.3 Building Redboot from Source Code

Step 1. Uncompress the Redboot source code:
% ***cd /home/sitsang***
% ***tar zxvf /tmp/RedBoot_Intel.tgz***
% ***cd RedBoot_Intel/***

Step 2. Edit the script file named 'linux_bld_redboot' to make the paths match user's installation. For example, locate the line:

REDBOOT_SRC=/home/alvin/workcvs/RedBoot_Intel/src

and change the path to match user's installation as follows:

REDBOOT_SRC=/home/sitsang/RedBoot-Intel/src

**Repeat this step for all of the absolute paths. The relative paths need not be changed**.

Step 3. Configure the redboot build option:

Locate the section of *cdl_option CYGBLD_GLOBAL_CFLAGS* in RedBoot_Intel/src/packages/hal/arm/cotulla/sitsang/current/cdl/hal_arm_sitsang.cdl.

For two flash banks version, make sure default_value is:
*default_value { "-Wall -Wpointer-arith -Wstrict-prototypes -Winline -Wundef -Woverloaded-virtual -g -O2 -fno-rtti -fno-exceptions -fvtable-gc -finit-priority -mapcs-frame" }*

For using only one flash bank, default_value should be:
*default_value { "**-DSITSANG_FLASH_SINGLE_BANK** -Wall -Wpointer-arith -Wstrict-prototypes -Winline -Wundef -Woverloaded-virtual -g -O2 -fno-rtti -fno-exceptions -fvtable-gc -finit-priority -mapcs-frame" }*

Step 4. % ***./linux_bld_redboot***

Run the build to completion. There are some warnings caused by some source files which intel never modified; These warnings can be safely ignored. The resulting binary file is 'redboot.bin' which is stored in:

./release-redboot/install/bin/redboot.bin

The binary image redboot.bin is in absolute binary format that is suitable for flashing via the JFlash utility, or burning with an off-board ROM burner to memory location 0x0.

## 3.4 Building Linux Kernel from Source Code

Step 1. Untar the Linux kernel source (downloaded from ftp.kernel.org):
% ***cd /home/sitsang/***

% ***tar xvzf /tmp/linux-2.4.19.tar.gz***

If the Linux kernel source was compressed with bzip2, use the following command to unzip it:
% ***bzip2 -dc /tmp/linux-2.4.19.tar.bz2 | tar xvf –***

Step 2. Create symbol link:
% ***ln –s linux-2.4.19 linux***

Step 3. Apply the whole patch package:
% ***gzip –dc /tmp/patch-2.4.19-sitsang2.gz | patch -p0***
Then go to the next step to build linux kernel for Sitsang board.

Optionally, user can apply the three patches by himself:

a) Apply Russell's arm-linux patches:
% ***gzip -dc /tmp/patch-2.4.19-rmk4.gz | patch -p0***
b) Apply Nicolas' Lubbock Linux patches:
% ***gzip -dc /tmp/diff-2.4.19-rmk4-pxa2.gz | patch -p0***
e) Apply Sitsang board patches (Designed by Alvin) :
% ***cat /tmp/diff-2.4.19-rmk4-pxa2-alvin2 | patch –p0***

Step 4. Configure kernel:

% ***cd /home/sitsang/linux***
% ***make sitsang_config***
% ***make oldconfig***

Step 5. Select both flash banks scheme or single flash bank scheme:
% ***make menuconfig***
**Find configure option: Memory Technology Devices (MTD) → *Mapping drivers for chip access →Sitsang uses single flash bank***
For both flash banks scheme, leave it blank; for single flash bank scheme, select it.

Step 6. Build the kernel after patching:
% ***make dep***
% ***make zImage***

Build the kernel modules:
% ***make modules***

The result binary is *zImage* and is stored in:
./arch/arm/boot/zImage

There are some result modules after make modules:

| *Module Name* | *Description* | *Location in Sitsang File System* |
|---|---|---|
| usbdcore.o | Core support for USB device | ./drivers/usb/device/usbdcore.o |
| serial_fd.o | USB device serial function support | ./drivers/usb/device/serial_fd/serial_fd.o |
| pxa_bi.o | USB Device bus interface support for PXA255 | ./drivers/usb/device/bi/pxa_bi.o |
| bsd_comp.o | PPP BSD-Compression | ./drivers/net/bsd_comp.o |
| ppp_async.o | PPP support for async serial ports | ./drivers/net/ppp_async.o |
| ppp_deflate.o | PPP Deflate compression | ./drivers/net/ppp_deflate.o |
| usbserial.o | USB serial converter support | ./driver/usb/serial/usbserial.o |
| 8390.o | NE2000 Ethernet basic routines | ./driver/net/8390.o |
| pcnet_cs.o | NE2000 compatible PCMCIA support | ./driver/net/pcmcia/pcnet_cs.o |

**Table 14: Kernel Module List**

## 3.5 Creating A JFFS2 File System Image

User may use the following files to build a JFFS2 file image:
* sitsang_fs.tgz - root file tree
* mkfs.jffs2 - JFFS2 make file system utility

The mkfs.jffs2 binary executable utility included with this distribution has been tested on RedHat Linux v7.2 and Mandrake Linux v7.2.

First create a temporary subdirectory and expand the root file system into it.

```
% cd /home/sitsang/
% mkdir tmp
% cd tmp
% tar xvzf /tmp/sitsang_fs.tgz
% cd ..
% cp /tmp/mkfs.jffs2 .
% chmod   755  mkfs.jffs2
```

Then make user's change and run 'mkfs.jffs2' to create the new JFFS2 file image:

```
% ./mkfs.jffs2 -r /home/sitsang/tmp/sitsang_fs/  -o sitsang_fs_test.jffs2 -e 0x40000 --
pad=0x02800000
```

**NOTE 1:** The parameter '--pad=' specify the size of the file system image.

**NOTE 2:** '--pad=0x02800000' switch generates a padded 40mb image to avoid those pesky “JFFS2: Erase block.” warning messages at kernel boot up, and user can set other length value.

**NOTE 3:** For using both flash banks, the maximum size of the file system image can be 61M (0x03D00000); For using only one flash bank, the maximum size of the file system image can be 29M (0x01D00000).